

12장. 함수

12장. 함수

12.1 함수란?

1. 함수 : 자바스크립트에서 가장 중요한 핵심 개념
2. 함수 : 일련의 과정을 문으로 구현하고 코드 블록으로 감싸서 하나의 실행 단위로 정의한 것
3. 함수 내부로 입력 전달받는 변수 : 매개변수(parameter), 인수(argument), 반환값(return value)
4. 함수 정의(function definition)

```
// 함수 정의
function add(x, y){
    return x + y;
}
```

12.2 함수를 사용하는 이유

1. 함수는 몇 번이든 호출할 수 있으므로 **코드의 재사용**이라는 측면에서 매우 유용
2. **유지보수의 편의성**을 높이고, **코드의 신뢰성을 높이는 효과**가 있다.
3. 적절한 함수 이름은 내부 코드를 이해하지 않고도 함수의 역할을 파악할 수 있게 돕는다.
→ **코드의 가독성 향상**

12.3 함수 리터럴

1. 함수는 객체다.
2. 일반 객체는 호출할 수 없지만, 함수는 호출할 수 있다.

12.4 함수 정의

1. 변수는 '선언(declaration)' 한다고 했지만, 함수는 '정의(definition)'한다고 표현

12.4.1 함수 선언문

1. [예제 12-05]

```
// 함수 선언문
function add(x, y) {
    return x + y;
}

// 함수 참조
// console.dir은 console.log와는 달리 함수 객체의 프로퍼티까지 출력
// 단, Node.js 환경에서는 console.log와 같은 결과가 출력
console.dir(add); // f add(x, y)

// 함수 호출
console.log(add(2, 5)) // 7
```

2. 함수 선언문은 함수 이름을 생략할 수 없다.

12.4.2 함수 표현식

1. 값의 성질을 갖는 객체 : 일급 객체 (자바스크립트의 함수)

2. 함수 리터럴의 함수 이름은 생략할 수 있다 = 익명 함수

12.4.3 함수 생성 시점과 함수 호이스팅

1. 함수 호이스팅(function hoisting)

: 함수 선언문이 코드의 선두로 끌어 올려진 것처럼 동작하는 자바스크립트 고유의 특징